AFRL-RY-WP-TP-2008-1157

NONLINEAR FILTERING TECHNIQUES FOR GNSS DATA PROCESSING (PREPRINT)

Chun Yang, Ph.D. and Mikel Miller, Ph.D.

Sigtem Technology, Inc.

JUNE 2005

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
June 2005	Conference Paper Preprint	08 April 2005 – 08 June 2005

4. TITLE AND SUBTITLE
NONLINEAR FILTERING TECHNIQUES FOR GNSS DATA PROCESSING (PREPRINT)

5a. CONTRACT NUMBER
FA8650-05-C-1828

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
65502F

6. AUTHOR(S)
Chun Yang, Ph.D. (Sigtem Technology, Inc.)
Mikel Miller, Ph.D. (AFRL/RYRN)

5d. PROJECT NUMBER
3005

5e. TASK NUMBER
13

5f. WORK UNIT NUMBER
300513CY

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sigtem Technology, Inc.
1343 Parrott Drive
San Mateo, CA 94402-3630

Reference Systems Branch (AFRL/RYRN)
RF Sensor Technology Division
Air Force Research Laboratory, Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command, United States Air Force

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYRN

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TP-2008-1157

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Paper produced under contract FA8650-05-C-1828 for technical report AFRL-RY-WP-TR-2008-1137, SOFTWARE TOOLKIT FOR NONLINEAR FILTERS FOR GLOBAL POSITIONING SYSTEM (GPS) OPERATIONAL CONTROL SEGMENT (OCS) ESTIMATION AND OTHER APPLICATIONS.

Conference paper published in the Proceedings of the 61st Annual Meeting of The Institute of Navigation, 2005 (held Jun 27-29 2005, Cambridge, MA; Publisher: Institute of Navigation).

PAO Case Number: SN 05-0010; Clearance date: 18 Jan 2005. The U.S. Government is joint author of this work and has the right to use, modify, reproduce, release, perform, display, or disclose the work. Paper contains color.

14. ABSTRACT
Nonlinearities appear "everywhere" in the signal and data processing chain of the Global Navigation Satellite System (GNSS). At the "upper" end of the chain, ephemeris data modulated onto transmitted signals are predicted from satellite orbits whose determination is a well-known nonlinear estimation problem. At the "lower" end, within a GNSS receiver, the satellite signal tracking, position-fixing, and even integration with other sensors, such as an inertial navigation system (INS), all involve nonlinearity issues in one form or another. Either a small signal model or linearization is presently used to deal with nonlinearity. The former includes code and carrier tracking loops and the latter includes the extended Kalman filter (EKF) for orbit determination, position solution, and GPS/INS integration among others.
Abstract concludes on reverse side →

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Thao Nguyen
Unclassified	Unclassified	Unclassified	SAR	20	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

14. ABSTRACT (concluded)

In this paper, we present two emerging nonlinear filtering techniques, namely, the unscented Kalman filter (UKF) and particle filter (PF), and study their use in GNSS applications in comparison to the EKF. The UKF is also called the sigma-point Kalman filter (SPKF) and the PF has many variants in its implementation. In the EKF, both the state dynamics and measurement equations are linearized in order to apply the Kalman filter, which is only valid for linear Gaussian systems. Instead of truncating the nonlinear functions to the first order as in the EKF, the UKF and PF approximate the distribution of the state deterministically (sigma points) and randomly (particles), respectively, with a finite set of samples, and then propagate these points or particles through the exact nonlinear functions. Because the nonlinear functions are used without approximation, it is much simpler to implement and generates better results.

After formulating these nonlinear filtering algorithms, this paper will illustrate their functionality and performance using satellite orbit determination as an example via computer simulation. Furthermore, we will discuss implementation issues and analyze the use of these nonlinear filtering techniques to solve other nonlinear problems in GNSS and navigation applications.

Nonlinear Filtering Techniques for GNSS Data Processing

Chun Yang, Ph.D.
Sigtem Technology, Inc.

Mikel Miller, Ph.D.
AFRL/RYRP

BIOGRAPHIES

Dr. Chun Yang received his title of Docteur en Science from the Université de Paris (No. XI, Orsay), France, in 1989. After two years of post-doctoral research at the University of Connecticut, he moved on with his industrial R&D career. Since 1993, he has been with Sigtem Technology, Inc. and has been working on numerous GPS, integrated inertial, and adaptive array related projects.

Dr. Mikel M. Miller received his Ph.D. in Electrical Engineering from the Air Force Institute of Technology (AFIT), WPAFB, Ohio, in 1998. Up to 2003, he was an Assistant Professor of Electrical Engineering at AFIT. Dr. Miller is a Senior Electronics Engineer in the Reference Sensors and Receiver Applications Branch, Sensors Directorate, Air Force Research Laboratory (AFRL), WPAFB, Ohio, is also an Adjunct Professor of Electrical Engineering at AFIT and Miami University of Ohio. Dr. Miller has been involved with navigation-related research, development, and test since 1989 and his current areas of interest include inertial navigation system (INS) and Global Positioning System (GPS) receiver integration, advanced topics in Kalman filtering, autonomous vehicle navigation and control, multi-sensor fusion, and personal navigation and physiological monitoring.

ABSTRACT

Nonlinearities appear “everywhere” in the signal and data processing chain of the Global Navigation Satellite System (GNSS). At the “upper” end of the chain, ephemeris data modulated onto transmitted signals are predicted from satellite orbits whose determination is a well-known nonlinear estimation problem. At the “lower” end, within a GNSS receiver, the satellite signal tracking, position-fixing, and even integration with other sensors, such as an inertial navigation system (INS), all involve nonlinearity issues in one form or another. Either a small signal model or linearization is presently used to deal with nonlinearity. The former includes code and carrier tracking loops and the latter includes the extended Kalman filter (EKF) for orbit determination, position solution, and GPS/INS integration among others.

In this paper, we present two emerging nonlinear filtering techniques, namely, the unscented Kalman filter (UKF) and particle filter (PF), and study their use in GNSS applications in comparison to the EKF. The UKF is also called the sigma-point Kalman filter (SPKF) and the PF has many variants in its implementation. In the EKF, both the state dynamics and measurement equations are linearized in order to apply the Kalman filter, which is only valid for linear Gaussian systems. Instead of truncating the nonlinear functions to the first order as in the EKF, the UKF and PF approximate the distribution of the state deterministically (sigma points) and randomly (particles), respectively, with a finite set of samples, and then propagate these points or particles through the exact nonlinear functions. Because the nonlinear functions are used without approximation, it is much simpler to implement and generates better results.

After formulating these nonlinear filtering algorithms, this paper will illustrate their functionality and performance using satellite orbit determination as an example via computer simulation. Furthermore, we will discuss implementation issues and analyze the use of these nonlinear filtering techniques to solve other nonlinear problems in GNSS and navigation applications.

INTRODUCTION

Kalman filtering is widely used in both GPS Control Segment and User Segment. The GPS user solution is

calculated by a Kalman filter embedded in the GPS receiver, which makes use of the Space Vehicle (SV) orbit and time scale information contained in the navigation data sets generated by the Operational Control Segment (OCS) again using a Kalman filter.

It is well known that the Kalman filter was developed based upon linear dynamic and measurement models with additive white Gaussian noises. Under these assumptions, the state and its propagation remain Gaussian and are completely characterized by the probability density functions in terms of the mathematical expectation (i.e., mean value) and covariance matrix. Indeed, the Kalman filter provides the recursive equations to compute the mean value (i.e., the state estimate) and the estimation error covariance for propagation over time and for updating when the state measurements are available.

For many practical applications, however, the state dynamic and/or measurement equations are not linear. In order to apply the Kalman filter, conventional approaches would linearize the nonlinear dynamic and measurement equations using Taylor series expansion (usually to the first order but sometimes to the second order terms) either around the predicted state values or along a pre-calculated reference trajectory, resulting in the so-called extended Kalman filter (EKF) [Maybeck, 1982]. This EKF approach has been used successfully in many applications but its implementation requires extensive tuning, continuous monitoring, and on-line adjustment if necessary to avoid bias and instability.

It has been shown that it is much easier to approximate a Gaussian distribution than it is to approximate arbitrary nonlinear functions. Based on this, the unscented Kalman filter (UKF) has recently been introduced [Julier, Uhlman and Durrant-Whyte, 1995; Julier, 2002; Julier and Uhlman, 2004], which produces more accurate results than the EKF. Instead of truncating nonlinear functions to first order as in the EKF, the UKF approximates the distribution of the state with a finite set of samples (i.e., deterministically selected from the Gaussian distribution). These samples, called sigma points, are then propagated through the true nonlinear functions. The mean and covariance of the distribution after propagation and update are calculated as the weighted sum of the propagated sigma points and the weighted sum of their outer products, respectively. Because the nonlinear functions are used without approximation nor derivatives, it is much simpler to implement and generates better results (the performance is superior or similar to a truncated second order EKF but without the need to calculate Jacobians or Hessians of the nonlinear functions). However, the UKF cannot be applied to general non-Gaussian distributions.

Sequential Monte Carlo methods or particle filters [Doucet, de Freitas, and Gordon, 2000; Liu 2001] allow for a complete representation of the posterior distribution of the states so that any statistical estimates such as means, modes, kurtosis and variances can be computed, thus providing a tool to deal with any nonlinearities or distributions. The basis of particle filters is the sequential importance sampling (SIS) algorithm, which is also known as bootstrap filtering, interacting particle approximation, and survival of the fittest among others. The key idea is to represent the posterior density function of the state given measurements by a set of random samples with associated weights and to compute the estimates based on the samples and weights. This point mass (hence the name “particle”) representation of a probability density generalizes the traditional Kalman filtering method to nonlinear, non-Gaussian, on-line estimation [Arulampalam et al., 2002].

The crucial step in the design of a particle filter is therefore how to choose the importance function or proposal distribution that can approximate the desired unknown posterior distribution reasonably well. The most common choice is to sample from the probabilistic model of the state evolution (transition prior) but it may fail if measurements appear in the tail of the prior or if the likelihood function is too peaked in comparison to the prior. To overcome this problem, the EKF or UKF-based Gaussian approximation can be used as the proposal distribution [Haykin 2001]. The latter also allows the control of the rate at which the tails of the proposal importance distribution go to zero, thus generating proposal distribution with larger high order moments and means that are closer to the true mean of the target distribution [van de Merwe, 2004].

In this paper, we are interested in these emerging nonlinear filtering techniques for GNSS related applications. In principle, they can be used wherever the EKF is used today in GNSS and other navigation applications. This includes orbit determination to be illustrated in this paper, UKF or PF-based position fixing with pseudoranges in GPS receiver, carrier phase tracking with PF [Amblard, Brossier, and Moisan, 2002], integer ambiguity resolution with PF [Azimi-Sadjadi and Krishnaprasad, 2001], blind equalization and estimation in fading channels [Djuric et al., 2003], and GPS/INS integration with PF [Frykman, 2003]. The EKF used in joint multipath estimation and PRN code acquisition [Iltis, 1990] can be replaced by a UKF or PF. Other applications in navigation include quaternion-based orientation tracking [Kraft, 2003], cellular phone measurements-based positioning, map matching to digital elevation profile, and collision avoidance for cars [Gustafsson et al., 2003; Pham, Dahia, and Musso, 2003].

After formulating the nonlinear filtering algorithms, we will discuss their implementation issues in this paper. We will present computer simulation results to illustrate the

functionality and performance of the nonlinear filters using satellite orbit determination as an example.

NONLINEAR FILTERING WITH EKF, UKF & PF

In this section, some equivalent models that are suitable for nonlinear filter development are first described for nonlinear dynamic and measurement equations. The UKF/SPKF and PF are then presented. For simplicity, the details for the EKF are omitted, which can be found in many textbooks such as [Maybeck, 1982].

Nonlinear Dynamic and Measurement Models

A nonlinear dynamic system can be represented by a state transition model with at least three forms, namely, the probability density, regression, and first-order linearization:

$$p(\underline{x}_t|\underline{x}_{t-1}) \tag{1a}$$
$$\underline{x}_t = \underline{f}(\underline{u}_{t-1}, \underline{x}_{t-1}, \underline{v}_{t-1}) \tag{1b}$$
$$\underline{x}_t \approx \underline{f}(\underline{u}_{t-1}, \bar{\underline{x}}_{t-1}, \bar{\underline{v}}) + F_t(\underline{x}_{t-1} - \bar{\underline{x}}_{t-1}) + G_t(\underline{v}_{t-1} - \bar{\underline{v}}) \tag{1c}$$

where $\underline{x}_t \in R^{nx}$ denotes the state (including the hidden variables and/or parameters) of the system at time t, $\underline{u}_t \in R^{nu}$ is some known input, $\underline{v}_t \in R^{nv}$ is a noise term with covariance matrix Q, $\underline{f}$: $R^{nx} \times R^{nu} \times R^{nv} \rightarrow R^{nx}$ is the deterministic nonlinear state transition model, and $F_t = \frac{\partial f(x_t)}{\partial x_t}|_{x_t=\bar{x}_t}$ and $G_t = \frac{\partial f(v_t)}{\partial v_t}|_{v_t=\bar{v}}$ are the Jacobians of the state model linearized around the reference value $\bar{\underline{x}}_t$ and $\bar{\underline{v}}$, respectively.

By the same token, the nonlinear measurement model can be put into one of the three forms as:

$$p(\underline{y}_t|\underline{x}_t) \tag{2a}$$
$$\underline{y}_t = \underline{h}(\underline{u}_t, \underline{x}_t, \underline{n}_t) \tag{2b}$$
$$\underline{y}_t \approx \underline{h}(\underline{u}_t, \bar{\underline{x}}_t, \bar{\underline{n}}) + H_t(\underline{x}_t - \bar{\underline{x}}_t) + L_t(\underline{n}_t - \bar{\underline{n}}) \tag{2c}$$

where $\underline{y}_t \in R^{ny}$ is the measurement vector, $\underline{n}_t \in R^{nn}$ the measurement noise with covariance matrix R, which is assumed to be independent of $\underline{v}_t$, $\underline{h}$: $R^{nx} \times R^{nu} \times R^{nn} \rightarrow R^{nu}$ is the deterministic nonlinear measurement model, and $H_t = \frac{\partial h(x_t)}{\partial x_t}|_{x_t=\bar{x}_t}$ and $L_t = \frac{\partial h(n_t)}{\partial n_t}|_{n_t=\bar{n}}$ are the Jacobians of the measurement model linearized around the reference values $\bar{\underline{x}}_t$ and $\bar{\underline{n}}$, respectively.

UKF/SPKF

The EKF only uses the first order terms of the Taylor series when expanding the nonlinear functions. As such, the linearization validity must be checked to avoid large errors built up in the estimated posterior distribution of the state when the higher order terms of the Taylor series expansion become significant. For this reason, higher-order EKF and iterated EKF have been proposed in the past.

Unlike the EKF, the UKF does not approximate the nonlinear dynamic and measurement models; it utilizes the true nonlinear models but instead approximates the distribution of the state. When the state distribution is Gaussian, a minimal set of deterministically chosen samples (i.e., sigma points) can be used to completely capture the true mean value and covariance of the state. These sigma points then propagate through the true nonlinear system and again capture the posterior mean and covariance accurately to the second order for any nonlinearity, with errors only introduced in the third and higher orders. In comparison, the EKF only calculates the posterior mean and covariance accurately to the first order with all high order moments truncated.

The UKF is the application of the scaled unscented transformation (SUT) [Julier, 2002] to recursive minimum mean-square-error (MMSE) estimation. Consider a random variable $\underline{x} \in R^n \sim \mathcal{N}(\underline{m}_x, P_x)$. The covariance matrix is scaled up by a factor $(n+\kappa)$, thus spreading the standard deviation by a factor of $\gamma = \sqrt{n+\kappa}$. Now factorize the covariance matrix by, say, Cholesky decomposition, producing the square root $\sqrt{P_x}$. Construct 2n+1 sigma points around the mean vector $\underline{m}_x$ with the scaled spread square root $\gamma\sqrt{P_x}$ as:

$$\mathcal{X} = [\underline{m}_x, \underline{m}_x + \gamma\sqrt{P_x}, \underline{m}_x - \gamma\sqrt{P_x}] \tag{3}$$

Let $[\mathcal{X}]_i$ be the i^{th} column of $\mathcal{X}$ (i.e., the i^{th} sigma point vector, with the count starting from 0). Propagating it through a nonlinear function $\underline{y} = f(\underline{x})$ produces the posterior sigma point vectors as $[\mathcal{Y}]_i = f([\mathcal{X}]_i)$ for i = 0 … 2n. The mean and covariance are calculated from the weighted sample mean and covariance as:

$$\underline{m}_y = \sum_{i=0}^{2n} W_i^m [\mathcal{Y}]_i \tag{4a}$$
$$P_{yy} = \sum_{i=0}^{2n} W_i^c ([\mathcal{Y}]_i - \underline{m}_y)([\mathcal{Y}]_i - \underline{m}_y)^T \tag{4b}$$
$$P_{xy} = \sum_{i=0}^{2n} W_i^c ([\mathcal{X}]_i - \underline{m}_x)([\mathcal{Y}]_i - \underline{m}_y)^T \tag{4c}$$

where W_i^m and W_i^c are the constant weights for the mean and covariance, respectively, given by:

$$W_0^m = \frac{\lambda}{n+\lambda}, \; W_i^m = \frac{1}{2(n+\lambda)}, \; i = 1, \ldots, 2n \tag{5a}$$
$$W_0^c = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta), \; W_i^c = W_i^m, \; i = 1, \ldots, 2n \tag{5b}$$

where $\lambda = \alpha^2(n+\kappa)$.

The parameter $\kappa \geq 0$ is chosen to ensure the positive semi-definiteness of the covariance matrix, which defaults to zero. The parameter $0 \leq \alpha \leq 1$ controls the size of the sigma point distribution (probabilistic spread in terms of covariance) and takes a small number to avoid sampling non-local effects when nonlinearities are strong. The

parameter β is a weighting term used to incorporate any knowledge of higher order moments of the distribution. For a Gaussian, the optimal choice β = 2 [van de Merwe, 2004].

Figure 1 shows the propagation of a random variable through a nonlinear function with the SUT (as in the UKF) in comparison to the first order linearization (as in the EKF). Its application to recursive estimation leads to the UKF, which includes the following steps. Step 1 – Construct an augmented state by concatenating the original state with the process noise and measurement noise. Step 2 – Calculate the covariance matrix for the augmented state and select the sigma points for the augmented state. Step 3 – Conduct the time-update by propagating the sigma points through the exact nonlinear process and measurement equations. Step 4 – Calculate the covariance matrix of the predicted measurements per sigma points (P_{YY}) and the cross-covariance matrix between the predicted measurements and the augmented state (P_{XY}). Step 5 – Conduct the measurement update using the Kalman filter gain as $\mathcal{K}_k = P_{YY}(P_{XY})^{-1}$.

Table 1 lists the algorithm that updates the mean and covariance of the Gaussian approximation to the posterior distribution of the state. As shown, there is no explicit calculation of Jacobians or Hessians. Since the covariance matrix can be expressed recursively in the square-root form, not only does the UKF outperform the EKF in accuracy and robustness, it does so at no extra computational cost. When the process and measurement noise terms are purely additive, there is no need to augment the system state with noise, further reducing the computational complexity.

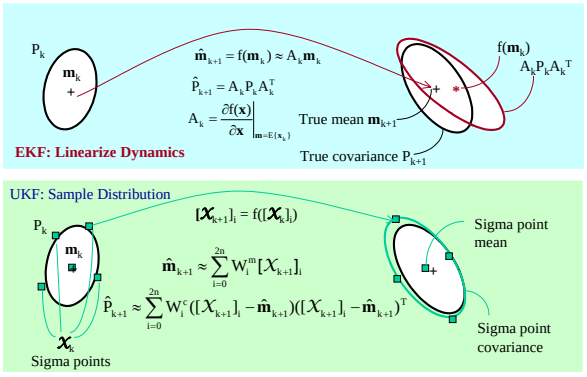

Figure 1 – Propagation of Random Variable Through Nonlinear Function: UKF vs. EKF

Particle Filter and Hybrid Implementation

Both the EKF and UKF reply on the Gaussian assumption and its approximation. The particle filtering method described in this section, however, does not require this assumption. But it has problems of its own. To overcome such problems, the particle filter may be combined with an EKF or UKF in a hybrid manner. The design of a particle filter is based on four basic ideas, which are discussed below.

Discrete Approximation of PDF

The first idea is the approximation of a continuous-support distribution $p(x_{0:t}|y_{1:t})$ by N discrete samples $x_{0:t}^{(i)}$, "randomly" drawn from the distribution $p(x_{0:t}|y_{1:t})$, for i = 1, …, N:

$$p(x_{0:t} \mid y_{1:t}) \cong \frac{1}{N}\sum_{i=1}^{N}\delta(x_{0:t} - x_{0:t}^{(i)}) \tag{6}$$

where the subscript "0:t" or "1:t" indicates the observation interval from 0 or 1 to t and δ(•) is the Dirac delta function. This is the so-called Monte Carlo method. With this approximation, the computation of expectation of any function of $x_{0:t}$, g(•), is reduced from a complicated integration to a simple summation as:

$$E\{g_t(x_{0:t})\} = \int g_t(x_{0:t})p(x_{0:t} \mid y_{1:t})dx_{0:t} \cong \frac{1}{N}\sum_{i=1}^{N} g_t(x_{0:t}^{(i)}) \tag{7}$$

Concatenation:	Augment the state vector to include noise terms: $\underline{x}_t^a = [\underline{x}_t^T\ \underline{v}_t^T\ \underline{n}_t^T]^T$, $n_a = n_x + n_v + n_n$
Initialization:	$\bar{\underline{x}}_0 = E\{\underline{x}_0\}$, $P_0 = E\{(\underline{x}_0 - \bar{\underline{x}}_0)(\underline{x}_0 - \bar{\underline{x}}_0)^T\}$ $\bar{\underline{x}}_0^a = E\{\underline{x}_0^a\} = [\bar{\underline{x}}_0^a, 0, 0]$, $P_0^a = E\{(\underline{x}_0^a - \bar{\underline{x}}_0^a)(\underline{x}_0^a - \bar{\underline{x}}_0^a)^T\} = diag\{P_0, Q, R\}$
Recursive over t:	
Calculate Sigma Points	$\mathcal{X}_{t-1}^a = [\mathcal{X}_{0,t-1}^a, \mathcal{X}_{1,t-1}^a, \ldots, \mathcal{X}_{2n_a,t-1}^a]$ where $\mathcal{X}_{0,t-1}^a = \bar{\underline{x}}_{t-1}^a$, $W_0^{(m)} = \lambda/(n_x+\lambda)$, $W_0^{(c)} = \lambda/(n_x+\lambda) + (1-\alpha^2+\beta)$ $\mathcal{X}_{i,t-1}^a = \bar{\underline{x}}_{t-1}^a + [\sqrt{(n_a+\kappa)P_{t-1}^a}\,]_i$, $i = 1, \ldots, n_a$ $\mathcal{X}_{i,t-1}^a = \bar{\underline{x}}_{t-1}^a - [\sqrt{(n_a+\kappa)P_{t-1}^a}\,]_i$, $i = n_a+1, \ldots, 2n_a$ $W_i^{(m)} = W_i^{(c)} = 1/(2(n_x+\lambda))$, $i = 1, \ldots, 2n_a$
Time Update	$\mathcal{X}_{t|t-1}^x = f(\underline{u}_{t-1}, \mathcal{X}_{t-1}^x, \mathcal{X}_{t-1}^v)$ $\bar{\underline{x}}_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(m)}\mathcal{X}_{i,t|t-1}^x$ $P_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(c)}[\mathcal{X}_{i,t|t-1}^x - \bar{\underline{x}}_{t|t-1}][\mathcal{X}_{i,t|t-1}^x - \bar{\underline{x}}_{t|t-1}]^T$ $\underline{\mathcal{Y}}_{t|t-1} = \underline{h}(\underline{u}_{t-1}, \mathcal{X}_{t|t-1}^x, \mathcal{X}_{t-1}^n)$ $\bar{\underline{y}}_{t|t-1} = \sum_{i=0}^{2n_a} W_i^{(m)}\underline{\mathcal{Y}}_{i,t|t-1}$
Measurement Update	$P_{\tilde{y}_t\tilde{y}_t} = \sum_{i=0}^{2n_a} W_i^{(c)}[\underline{\mathcal{Y}}_{i,t|t-1} - \bar{\underline{y}}_{t|t-1}][\underline{\mathcal{Y}}_{i,t|t-1} - \bar{\underline{y}}_{t|t-1}]^T$ $P_{x_t y_t} = \sum_{i=0}^{2n_a} W_i^{(c)}[\mathcal{X}_{i,t|t-1} - \bar{\underline{x}}_{t|t-1}][\underline{\mathcal{Y}}_{i,t|t-1} - \bar{\underline{y}}_{t|t-1}]^T$ $\mathcal{K}_t = P_{x_t y_t}(P_{\tilde{y}_t\tilde{y}_t})^{-1}$ $\bar{\underline{x}}_t = \bar{\underline{x}}_{t|t-1} + \mathcal{K}_t(\underline{y}_t - \bar{\underline{y}}_{t|t-1})$ $P_t = P_{t|t-1} - \mathcal{K}_t P_{\tilde{y}_t\tilde{y}_t}\mathcal{K}_t^T$

Table 1 - Unscented Kalman Filter

Importance Sampling

The second idea is the importance sampling. In the estimation problem, the posterior distribution $p(x_{0:t}|y_{1:t})$ is in fact what we want to estimation from data, thus not available for sampling directly. One way to get around this is to approximate the expectation over the unknown distribution $p(x_{0:t}|y_{1:t})$ by another expectation taken over a

known easy-to-sample distribution $q(x_{0:t}|y_{1:t})$, called importance function , also known as proposal distribution (which we will use interchangeably):

$$E\{g_t(x_{0t})\} = \int g_t(x_{0t}) \frac{p(x_{0t} \mid y_{1t})}{q(x_{0t} \mid y_{1t})} q(x_{0t} \mid y_{1t}) dx_{0t}$$

$$= \int g_t(x_{0t}) \frac{p(y_{1t} \mid x_{0t}) p(x_{0t}) / p(y_{1t})}{q(x_{0t} \mid y_{1t})} q(x_{0t} \mid y_{1t}) dx_{0t}$$

$$= \int g_t(x_{0t}) \frac{w_t(x_{0t})}{p(y_{1t})} q(x_{0t} \mid y_{1t}) dx_{0t} \quad (8)$$

where the variable $w_t(x_{0:t})$ is the unnormalized importance weight defined by:

$$w_t(x_{0:t}) = \frac{p(y_{1t} \mid x_{0t}) p(x_{0t})}{q(x_{0t} \mid y_{1t})} \quad (9)$$

The probability density function $p(y_{1:t})$ in the denominator of Eq. (8) can be evaluated as:

$$p(y_{1t}) = \int p(y_{1t} \mid x_{0t}) p(x_{0t}) \frac{q(x_{0t} \mid y_{1t})}{q(x_{0t} \mid y_{1t})} dx_{0t}$$

$$= \int w_t(x_{0t}) q(x_{0t} \mid y_{1t}) dx_{0t} = E_q\{w_t(x_{0t})\} \quad (10)$$

where $E_q\{\bullet\}$ is the expectation taken over the importance function or proposal distribution $q(x_{0:t}|y_{1:t})$.

Following Eq. (10), $p(y_{1:t}) = E_q\{w_t(x_{0:t})\}$ can be taken out of the integral of Eq. (8), still in the denominator, though. By consequence, the numerator of Eq. (8) can be evaluated over $q(x_{0:t}|y_{1:t})$ in the same manner, leading to $E_q\{g_t(x_{0:t})w_t(x_{0:t})\}$.

When these expectations are approximated as in Eq. (6) by drawing N samples from the importance function (proposal distribution) $q(x_{0:t}|y_{1:t})$, we have:

$$E\{g_t(x_{0t})\} = \frac{E_q\{w_t(x_{0t}) g_t(x_{0t})\}}{E_q\{w_t(x_{0t})\}} \approx \frac{1/N \sum_{i=1}^{N} g_t(x_{0t}^{(i)}) w_t(x_t^{(i)})}{1/N \sum_{i=1}^{N} w_t(x_t^{(i)})}$$

$$= \sum_{i=1}^{N} g_t(x_{0t}^{(i)}) \frac{1/N\, w_t(x_t^{(i)})}{1/N \sum_{j=1}^{N} w_t(x_t^{(j)})}$$

$$= \sum_{i=1}^{N} g_t(x_{0t}^{(i)}) \tilde{w}_t(x_t^{(i)}) \quad (11a)$$

$$\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} \quad (11b)$$

The normalized importance weight $\tilde{w}_t^{(i)}$ is used in the last equality of Eq. (11a) with $\sum_{i=1}^{N} \tilde{w}_t^{(i)} = 1$.

Sequential Importance Sampling (SIS)

The third idea is the sequential importance sampling (SIS). Under the assumptions that the underlying state corresponds to a Markov process (i.e., $p(x_{0:t}) = p(x_0)\Pi_{j=1}^{t} p(x_j|x_{j-1})$), the observations are conditionally independent given the state (i.e., $p(y_{1:t}|x_{0:t}) = \Pi_{j=1}^{t} p(y_j|x_j)$), and the importance function or proposal distribution is factorable (i.e., $q(x_{0:t}|y_{1:t}) = q(x_{0:t-1}|y_{1:t-1}) q(x_t|x_{0:t-1}, y_{1:t-1})$ according to the Bayes' rule), then the unnormalized importance weight can be estimated recursively as:

$$w_t(x_{0t}) = \frac{(p(y_{1t-1} \mid x_{0t-1}) p(y_t \mid x_t))(p(x_{0t-1}) p(x_t \mid x_{t-1}))}{q(x_{0t-1} \mid y_{1t-1}) q(x_t \mid x_{0t-1}, y_{1t})}$$

$$= w_{t-1}(x_{0t-1}) \frac{p(y_t \mid x_t) p(x_t \mid x_{t-1})}{q(x_t \mid x_{0t-1}, y_{1t})} \quad (12)$$

Eq. (12) provides a mechanism to sequentially update the importance weight given the conditional proposal distribution $q(x_t|x_{0:t-1}, y_{1:t})$. Indeed, we can sample from this proposal distribution (i.e., generate N discrete samples $x^{(i)}_t$ according to $q(x_t|x_{0:t-1}, y_{1:t})$) and evaluate the likelihood and transition probabilities (i.e., $p(y_t|x_t)$ and $p(x_t|x_{t-1})$ given by the process and measurement models of Eqs. (1a) and (2a), respectively) for theses samples. Table 2 lists the algorithm for a generic particle filter with additional steps discussed below.

Table 2. Generic Particle Filter

Initialization: t = 0	Draw the initial state $x_0^{(i)}$ from the prior $p(x_0)$, $i = 1, \ldots, N$
For t = 1, 2, …	
Importance Sampling	Sample $\tilde{x}_t^{(i)} \sim q(x_t \mid x_{0t-1}^{(i)}, y_{1t})$ and set $\tilde{x}_{0t}^{(i)} = [\hat{x}_{0t-1}^{(i)}, \tilde{x}_t^{(i)}]$ for $i = 1, \ldots, N$
	Evaluate the unnormalized importance weight: $w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t \mid \tilde{x}_t) p(\tilde{x}_t \mid \hat{x}_{t-1})}{q(\tilde{x}_t^{(i)} \mid \hat{x}_{t-1}^{(i)}, y_{1t})}$ for $i = 1, \ldots, N$
	Normalize the importance weights: $\tilde{w}_t^{(i)} = w_t^{(i)} [\sum_{i=1}^{N} w_t^{(i)}]^{-1}$ for $i = 1, \ldots, N$
Selection (Resampling):	Multiply (or suppress) samples $\tilde{x}_t^{(i)}$ with high (or low) importance weights $\tilde{w}_t^{(i)}$ to obtain N random samples $x_{0t}^{(i)}$ approximately distributed according to $p(x_{0t}^{(i)} \mid y_{1:t})$
	Set $w_t^{(i)} = \tilde{w}_t^{(i)} = 1/N$ for i = 1, …, N
Output:	The output of the algorithm is a set of samples $\tilde{x}_{0t}^{(i)}$ with weights $\tilde{w}_t^{(i)} = 1/N$ for $i = 1, \ldots, N$
	The samples can be used to approximate the posterior distribution as $p(x_{0t} \mid y_{1t}) \cong \hat{p}(x_{0t} \mid y_{1t}) = \frac{1}{N} \sum_{i=1}^{N} \delta(x_{0t} - \tilde{x}_{0t}^{(i)})$
	The samples can also be used to calculate expectation for appropriate functions as: $E\{g_t(x_{0t})\} = \int g_t(x_{0t}) p(x_{0t} \mid y_{1t}) dx_{0t} \cong \frac{1}{N} \sum_{i=1}^{N} g_t(\tilde{x}_{0t}^{(i)})$ which is the marginal conditional mean of x_{0t} when $g_t(x_{0t}) = x_{0t}$ and the marginal conditional covariance of x_{0t} when $g_t(x_{0t}) = x_{0t} x_{0t}^T - E_{p(x_t \mid y_{1t})}\{x_{0t}\} E^T_{p(x_t \mid y_{1t})}\{x_{0t}\}$.

Resampling and Diversification

For any practical implementation, the number of samples that can be drawn from a distribution is limited. As such, the choice of importance function or proposal distribution becomes critical, creating other issues impeding the success of a particle filter.

An easy choice of importance function for a process model with additive Gaussian noise is the transition prior:

$$q(x_t|x_{0:t-1}, y_{1:t}) \approx p(x_t|x_{t-1}) \sim N\{f(u_{t-1}, x_{t-1}, 0), Q_{t-1}\} \quad (13)$$

However, this proposal distribution does not incorporate the latest data available and it runs the risk to deplete the samples in the sense that after a few iterations, one of the

normalized importance weights tends to 1 while the remaining weights tend to 0. This effectively removes a large number of samples from the sample set because their importance weights are numerically insignificant.

To avoid this degeneracy, the fourth idea is the importance resampling, also called selection, in which samples with low importance weights are eliminated while samples with high importance are multiplied, keeping the total population of samples in the same level. Techniques for resampling include sampling importance resampling (SIR), residual resampling, and minimum variance sampling.

Since the selection step favors the creation of multiple copies of the "fittest" particles (thus allows us to track the updated distributions), many "unfit" particles may end up with few or none copies, leading to sample impoverishment. To solve this problem, an additional step is therefore needed to introduce the sample diversification after the selection step without affecting the validity of the approximation. A brute force approach would increase the number of samples. But a refined technique is to implement a Markov chain Monte Carlo (MCMC) step, which moves new particles to areas of more interest in the state space by applying a Markov chain transition kernel [Ruanaidh and Fitzgerald, 1996]. Figure 2 illustrates this process.

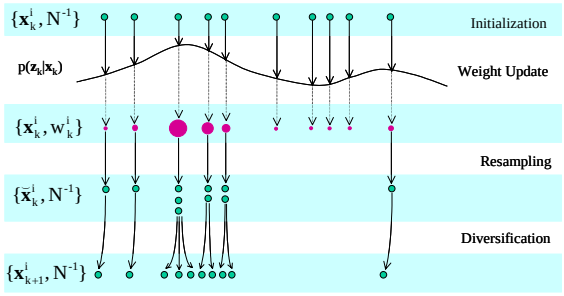

Figure 2- Resampling/Selection and Diversification

Hybrid Kalman Particle Filter to Move Proposal Density Closer to Likelihood Function

One cause for degeneracy frequently encountered when using importance functions as Eq. (13) is that the state transition prior and measurement likelihood density functions do not overlap (either too peaked or widely separated). This problem can be mitigated by moving particles to areas of high likelihood. One way to do so is to use a separate EKF to generate and propagate a Gaussian proposal distribution for each particle:

$$q(x_t^{(i)} \mid x_{0:t-1}^{(i)}, y_{1:t}) \approx N\{\bar{x}_t^{(i)}, \hat{P}_t^{(i)}\} \tag{14}$$

Since the UKF is more accurate than the EKF, thus having a more accurate importance function with bigger support overlap with the true posterior distribution, the estimates $\bar{x}_t^{(i)}$ and $\hat{P}_t^{(i)}$ from an UKF can be used to generate the proposal distribution as in Eq. (14). In this way, the EKF or UKF is effectively combined with a particle filter, yielding an extended Kalman particle filter (EKPF) or an unscented Kalman particle filter (UKPF).

It is clear that EKPF and UKPF require very high computational power particularly for systems of large dimensions. To strike a balance between performance and computation, the Gaussian mixture sigma point particle filter (GMSPPF) combines an importance sampling (IS) based measurement updating with a scaled unscented transformation (SUT) based Gaussian sum filter for the time updating and proposal density generation as shown in Figure 3.

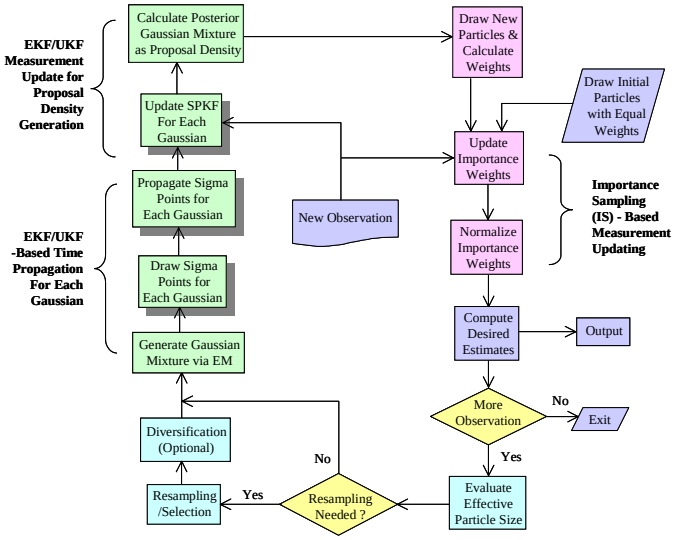

Figure 3 – Block Diagram of Hybrid Kalman Particle Filter

In GMSPPF, the IS-based measurement updating produces the weighted posterior particle set. It is followed by the resampling and diversification steps. The particles are then represented by a finite Gaussian mixture model found using the expectation-maximization (EM) algorithm. The resampling and diversification step may be omitted if a weighted EM algorithm is used in generating the Gaussian mixture thanks to its inherent kernel smoothing nature.

EKF or UKF/SPKF can be used to time-propagate and then measurement-update each particle or each component of the Gaussian mixture. The posterior Gaussian mixture is then used as the proposal density to draw particles. In this way, each measurement is used twice, the first time by an EKF or UKF/SPKF to bring the measurement-updated importance function closer to the

likelihood function and the second time by the importance sampling-based update to produce more accurate estimate of the posterior distribution. Other implementation schemes of particle filter can be found in [Doucet, de Freitas, Gordon 2001; Liu, 2001; Ristic, Arullampalam and Gordon, 2004].

SIMULATION SCENARIO AND FILTER MODELS

A simple tracking scenario is shown in Figure 4 where a satellite moves along a perfect circle of radius r_s = 26,560km at a constant angular velocity of $\omega_s = 360^o/12hr$ around the earth. The truth position, velocity, and acceleration components are generated by:

$$\theta(t) = \theta_0 - \omega t,\ \omega = 360/12\ (^o/hr) \quad (15a)$$

$$x(t) = r_s\cos(\theta(t)),\ y(t) = r_s\sin(\theta(t)) \quad (15b)$$

$$v_x(t) = r_s\omega\sin(\theta(t)),\ v_y(t) = -r_s\omega\cos(\theta(t)) \quad (15c)$$

$$a_x(t) = -\omega^2 x(t),\ a_y(t) = -\omega^2 y(t) \quad (15d)$$

With the notations $\ddot{\underline{r}} = \underline{a} = [a_x, a_y]'$ and $\underline{r} = [x, y]'$, Eq. (15d) can be written into the form of gravitational acceleration:

$$\ddot{\underline{r}} = -\omega^2 \underline{r} = -\frac{\mu}{r^3}\underline{r} \quad (16)$$

where $\mu = k^2M = 3{,}986\times10^8 m^3/s^2$ is the earth's gravitational constant, k^2 is the universal constant of gravitation, and M is the mass of the earth. The equation holds because $a = k^2M/r^2 = V^2/r = \omega^2 r$, which leads to $\omega^2 = a/r = \mu/r^3$.

Monitor stations on the surface of the earth with radius r_e = 6,378km are assumed to lie within the orbital plane, thus making it a 2D scenario. The local horizon established for the monitor station makes about 14^o with the x-axis in either direction, above which the satellite is visible to the monitor station. Since we will apply the kinematic method for orbit determination, the formulation is thus done in the earth-fixed coordinates.

The pseudorange measurement denoted by r for the monitor station located at (x_m, y_m) is related to the satellite position (x, y) in a nonlinear manner as:

$$r = \sqrt{(x-x_m)^2 + (y-y_m)^2} + n_r \quad (17)$$

where n_r is the measurement noise with variance σ_r^2.

The measurement noise variance is set to be $\sigma_r^2 = (0.1m)^2$ in the simulation, used by both the filters and in the measurement generation. Dual-frequency GPS receivers with P-code ranging have their pseudorange error variance around $(10m)^2$ whereas carrier phase ranging measurements have their error variance around $(0.01m)^2$ or better. This choice for our simulation is "arbitrary" for the sake of convenience.

The 2D satellite motion is considered by the tracker filter as two independent 3rd order models with the position, velocity, and acceleration as their state for each axis as:

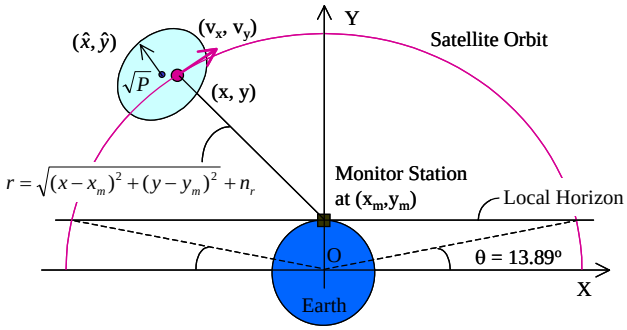

Figure 4 - Simple 2D Tracking Geometry

$$\begin{bmatrix} p \\ v \\ a \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ v \\ a \end{bmatrix}_k + \begin{bmatrix} \frac{T^2}{2} \\ T \\ 1 \end{bmatrix} w_k \quad (18)$$

where T = 1s is the sampling interval and the process noise $w_x(k)$ and $w_y(k)$ are random accelerations modeled as white Gaussian with their variances σ_{wx}^2 and σ_{wy}^2, respectively. In generating the truth orbit, we do not include any process noise (purely deterministic). However, the tracking filter will include a process noise with the variance $\sigma_{wx}^2 = \sigma_{wy}^2 = (0.1m/s^2)^2$ for the kinematics model of Eq. (18).

An orbiting satellite is under pulling force that changes all the time. For the orbit arc in Figure 4, v_x is always positive, changing from the initial value of 1,000m/s to the maximum of 3,874m/s and down to 1,000m/s again whereas v_y changes from near 4,000m/s to near -4,000m/s. The tangential linear velocity is a large number of 3,874m/s. The acceleration components a_x and a_y change about $0.5m/s^2$ over half of the two-hours arc. Over a short period of time, it can be considered constant provided that the process noise is properly adjusted to account for the changes. As such, the linear process model of Eq. (18) in the Cartesian coordinates is appropriate for orbit determination when accurate measurements are available at high sampling rate and but it does not provide any prediction capability.

Any tracking filter that is based on the model of Eq. (18) alone does not take advantage of the well-behaved motion pattern, which. Furthermore, if there is only one monitor station with range measurements, the problem of geometric dilution of precision (GDOP) appears. In our simulation, we therefore consider two monitor stations, which are located at $(x_m, y_m)) = (-r_e, 0)$ and $(0, r_e)$, respectively. The arc of satellite orbit that is visible to both stations is from 166^o to 104^o in the second quadrant.

SAMPLE BEHAVIORS AND ANALYSES

Four nonlinear filters, namely, EKF, UKF, PF, and GMSPPF, are implemented with sample behaviors presented here for visualization and comparison. In the simulation, the initial conditions are set as:

$$\hat{\underline{x}}_0 = \underline{x}_0 + [100, 10, 1, 100, 10, 1]^T \quad (19a)$$

$$P_0 = \text{diag}\{[100^2, 10^2, 1^2, 100^2, 10^2, 1^2]\} \quad (19b)$$

for all the nonlinear filters except for the particle filter for the reasons to be explained later.

Figures 6, 10, 14, and 18 show the true and estimated orbits for the four filters, EKF, UKF, PF, and GMSPPF, respectively. Except for the PF, the other three filter estimates are quite close with an approximate ranking of performance as EKF < UKF < GMSPPF.

Figures 7, 11, 15, and 19 show the true, measured, and estimated ranges (the top plot) and the range residual errors (the bottom plot) for the four filters as seen from monitor station 1. Figures 8, 12, 16, and 20 show the true, measured, and estimated ranges (the top plot) and the range residual errors (the bottom plot) for the four filters as seen from monitor station 2. Since the measurement noise standard deviation is set to be 0.1m, it is reasonable to see that the range residual errors are also around 0.1m.

Figures 9, 13, 17, and 21 show the estimation errors for position (top), velocity (middle) and acceleration (bottom) for the four filters. It can be seen that the estimates of the x-components are consistently better than the y-component counterpart over this short orbit arc because of the influence of GDOP.

To appreciate this, we run simulations over longer periods of time, which is increased from 100s in the previous simulations to the entire arc of 7466s (about 2 hrs). Only the results for UKF are shown here. Figure 22 shows the true and estimated orbit arc and Figure 23 shows the estimation errors for position (top), velocity (middle), and acceleration (bottom).

As the satellite moves away from the negative x-axis toward the positive y-axis (clockwise), the GDOP degrades for the x-component but improves for the y-component. This is visible in the top plot of Figure 23 as the x position error grows while the y position error is reduced. The position error ratio changes from $\Delta x:\Delta y$ = 1:4.5 to 4.5:1, consistent with what is shown in Figure 24 for the GDOP calculated as:

$$[\text{DOP}_x, \text{DOP}_y] = \text{diag}\{\sqrt{(H'H)^{-1}}\} \quad (20a)$$

$$H = [\frac{\partial h(\underline{x})}{\underline{x}}]_{x,y} = \begin{bmatrix} (x-x_1)/r_1 & (y-y_1)/r_1 \\ (x-x_2)/r_2 & (y-y_2)/r_2 \end{bmatrix} \quad (20b)$$

where (x_1, y_1) and (x_2, y_2) are the locations of the monitor stations with their ranges to the satellite being r_1 and r_2, respectively.

Figures 25 and 26 show the selection of initial sigma points for the UKF. The augmented state vector of the UKF includes the position (2), velocity (2), acceleration (2), process noise (2), and measurement noise (2). The total dimension is thus n = 10 and the number of sigma points is then 2n + 1 = 21.

Since we plot a subspace of dimension 2 at a time per figure, we can only show five distinctive sigma points per subspace, which are located at the center and two extremes of the major and minor axes, respectively, of the ellipse. The parameters used for the unscented transformation are $\alpha = 1$, $\beta = 2$, and $\kappa = 0$. The resulting radius is about 3σ, consistent with the spreading factor $\gamma = \sqrt{10} \approx 3$.

Figure 27 shows the initial 1,000 particles drawn from the initial distribution for the PF. Its processing results are presented in Figures 14 through 17. With 1,000 particles, the PF has to be initialized with much tighter conditions than the other three filters given in Eq. (19) as:

$$\hat{\underline{x}}_0 = \underline{x}_0 + [10, 0.1, 0.01, 10, 0.1, 0.01]' \quad (22a)$$

$$P_0 = \text{diag}\{[10^2, 0.1^2, 0.01^2, 10^2, 0.1^2, 0.01^2]\} \quad (22b)$$

Otherwise, the filter starts to diverge after few steps. The particular PF algorithm implemented in the simulation is the so-called bootstrap algorithm, also known as the condensation algorithm. As one of the rudimentary particle filters, it is known for such problems as sample impoverishment (depletion) among others.

In this PF algorithm, the proposal density is Eq. (13) and the particle weight updating is simplified into:

$$w^i_k = w^i_{k-1} \times p(\underline{z}_k | \underline{x}^i_{k|k-1})$$
$$= w^i_{k-1} \times \mathcal{N}\{\underline{z}_k - h(\underline{u}_k, \underline{x}^i_{k|k-1}, 0); R_k)\} \quad (23)$$

which is the product of the old weight and the likelihood value of the latest observation given the propagated particle. The problem stems from the calculation of likelihood values. In the EKF and UKF, the innovations (i.e., the measurement prediction errors), no matter how large it may be, are weighted by the Kalman filter gain determined by the product of the predicted state estimation error covariance (APA'+Q) and the inverse of the measurement error covariance $(HPH'+Q)^{-1}$.

For our measurement model with $\sigma_v^2 = (0.1\text{m})^2$, if the orbit position errors are larger than 10m, the measurement prediction errors (or the innovations) can be larger than 10m (further contributed by velocity and acceleration errors). This is equivalent to an error of $100\sigma_v$ in terms of the measurement noise standard deviation. The resulting likelihood values are practically zero.

If the initial state errors are large and few or none of the initial particles drawn are near the true state within 10m or closer, two phenomena have been observed. One is that

only one or few particles survive the updating while all the rest are rendered zero. The PF algorithm collapses because there are no sufficient particles to continue. The second is that all likelihood values are close to zero and practically the same and by consequence, they produce no meaningful changes to the weights, thus making the updating process ineffective (as if no measurement update).

To avoid the problem, one can either use smaller initial errors or a larger amount of particles. The latter may amount to excessive computation. We adopted the former technique in the present simulation with the small initialization errors as given by Eq. (22). Actually, various techniques have been proposed to perform on-line monitoring and editing [Gordon, Salmond, and Smith, 1993; Gordon, Salmond, and Ewing, 1995].

To illustrate, we now look at the resampling process of the particle filter at the 100^{th} time step. As shown in Figure 28, the top plot is the initial weights from the 99^{th} step in which all particles are assigned with an equal weight of 1/N with N = 1000 after resampling at the step.

At Step 100, N samples are first drawn from the distribution of the process noise. These noise samples are used together with the state samples (i.e., the particles of Step 99) to predict the state at Step 100 according to Eq. (18). The results are shown in the middle plot of Figure 28. Note that since the process noise variance is set to be very small $(0.1 m/s^2)^2$, what we see in Figure 28 is essentially the prediction driven by the velocity and acceleration estimates.

The bottom plot of Figure 28 shows the likelihood values for these predicted particles given an observation. It has a Gaussian shape.

The updated weights are shown in the top plot of Figure 29, which is the normalized product between the old weights and likelihood values. This also has a Gaussian shape. The middle plot of Figure 29 shows the state transition probability or the prior distribution of the state given the predicted state and estimated state of the previous step. In this particular simulation run, the resampling threshold is set to be N, a typical number used by the bootstrap algorithm. Since the effective number of particles is always less than N, resampling is done every step. The weights after resampling are again equal to 1/N, as shown in the bottom of Figure 29.

Figure 30 shows the particle indices before and after resampling. It has some discontinuities and horizontal segments. The latter indicates that those particles with large weights got multiplied during resampling. This is consistent with thegrouping seen in the top plot of Figure 29 for the weights prior to resampling.

To avoid depletion of particles or impoverishment, one technique is to redraw particles from an analytic distribution estimated from data as done in GMSPPF with a Gaussian mixture shown in Figures 18 to 21. Additional simulation results can be in [Yang, 2004].

MONTE CARLO SIMULATION RESULTS AND ANALYSES

The Monte Carlo simulation results are presented in this section. For easy comparison, we use the same simulation scenarios and the same initial conditions for the four nonlinear filters. While the initial conditions were held the same, the observation noise was different as it was drawn from the random noise generator in each Monte Carlo run. Similarly, the state particles were drawn differently for each run.

In the first case, we conducted 500 “independent” Monte Carlo runs with 1000 particles over 500 time steps (one second per step). Figure 32 shows the root mean square (RMS) errors of the measurement residuals at two monitor stations. The RMS value for an error is calculated using the following recursive formula:

$$RMS_e^2(n) = (1-1/n)\, RMS_e^2(n-1) + (1/n)e^2(n)$$
$$RMS_e(0) = 0,\ n = 1, 2, \ldots \qquad (24)$$

where e(n) is the error term of a variable of interest at the n^{th} Monte Carlo run and $RMS_e(n)$ is the resulting RMS value including that run.

In the following plots, we select not to show the results of the particle filter (the bootstrap algorithm) because it required a different initialization as explained previously. In addition, we observed some runs with divergence for the PF without applying any on-line editing as recommended in the original papers [Gordon, Salmond, and Smith, 1993; Gordon, Salmond, and Ewing, 1995] and this sort of instability never occurred to the other three filters. Otherwise, the RMS errors values for the PF are similar to the UKF and GMSPPF.

As shown in Figure 32, the RMS errors for the UKF and the GMSPPF are on the same level (better than 0.09 m) while the RMS errors for the EKF are on the order of 0.1 m. It is reasonable for the EKF because the measurement noise was set with a standard deviation of 0.1 m. However, it is somehow against intuition to see that the measurement residuals for the UKF and GMSPPF are slightly below the noise level. It remains to determine if this is because the filters “follow” the noise and/or have some variance reduction capability due to nonlinear averaging.

Figures 33 and 34 show the RMS values of the position estimation errors in X and Y, respectively. It can be seen that after the initial transient periods (due to large initial estimation error covariance and filter gain), the estimation

error RMS values remain consistent. The changes in position error RMS values are caused by GDOP (see Figure 24). In the X-dimension, the GDOP is initially close to 1 and grows up to 4.5 after 7500 s. For the first 500 steps (which started from 100s) as shown in Figure 33, the initial position error RMS is close to 0.1 m. When compared to the measurement error standard deviation, it represents a GDOP of 1, as predicted by the geometry.

In the Y-dimension, the GDOP is initially close to 4.5 and decreases down to 1 after 7500 s as shown in Figure 24. For the first 500 steps as shown in Figure 34 (which actually started from 100 s), the initial position error RMS is about 0.43 m for EKF and 0.35 m for UKF and GMSPPF. When compared to the measurement error standard deviation, it represents a GDOP of 4.3 and 3.5 for EKF and UKF/GMSPPF, respectively. The EKF, though producing larger errors, is closer to the value predicted by the geometry.

Figures 35 and 36 show the RMS values for the velocity errors in v_x and v_y, respectively. The RMS values were calculated over 95 Monte Carlo runs but during 7000 time steps (covering almost the half sky). The velocity error RMS values for UKF and GMSPPF change "monotonically" and are consistent with GDOP. That is, the v_x RMS grows up, as the X-GDOP gets worse. At the same time, the v_y RMS improves as the Y-GDOP becomes smaller.

However, the velocity error RMS values for EKF are more "dramatic" in the sense that the v_x RMS grows up exponentially rather than near-linearly with the X-GDOP. In addition, the v_y RMS exhibits a full cycle oscillation. This periodic behavior is somewhat correlated to the underlying velocity values. For the simulation considered, v_x is always positive whereas v_y changes sign at the mid-point.

We only considered about two hours of operation. When the simulation is done for the full orbit evolution, periodic behaviors are expected. Indeed, this has been observed in the GPS OCS Kalman filter with dynamics models [Brown, 1991]. For GPS, the state and measurement partials and the filter gains are periodic with the period being a sidereal day. The modeling errors in f(x) and h(x) are therefore repeatable per sidereal day assuming the same accuracy from day to day. As a result, the actual estimation errors induced by modeling errors in the functions f(x) and h(x) will become periodic.

In our simulation, the state model used by all filters is a linear kinematic model with constant velocity and acceleration per updating interval. Only the measurement equations are nonlinear. The EKF suffers from linearization errors while both the UKF and GMSPPF make use the nonlinear equations directly.

The GMSPPF has smaller RMS values than the UKF but they are so minuscule to be negligible. We also tested other cases. This included (1) different initial conditions with up to ten times errors and (2) large number of particles (up to 10000), which improved the results somewhat for the bootstrap filter but not very much for the GMSPPF (considerably slower though). The results remain similar.

In the GMSPPF implementation, we tried three to five Gaussian mixture components in each step to fit to the particle clouds. The initial mixture component parameters are obtained using the k-mean clustering algorithm. The actual fitting is done using the expectation-maximization (EM) algorithm. The iteration is stopped when the log likelihood values between two iterations drops below 0.001 or the number of iterations exceeds 10.

To illustrate, Figure 37 shows the result of one fitting example in a two-dimensional state space (i.e., position s vs. velocity $\dot{s}$) with 1000 particles. The resulting three mixture components have weights of 0.45064, 0.37583, and 0.17353, respectively. The 3σ ellipses are aligned almost with the axes, indicating small correlation between the two variables at the start of simulation.

Figure 38 shows another example of Gaussian mixture fitting with three components. Their respective weights are 0.40677, 0.39830, and 0.19493. In this case, the 3σ ellipses are rotated with respect to the axes, indicating large correlation between the two variables at the end of simulation.

It is reasonable to conclude at this point that (1) in the cases with Gaussian noise, both the UKF/SPKF and GMSPPF outperform the EKF and (2) the UKF/SPFK performs practically as good as the GMSPPF but with significantly less computation. It is therefore recommended to try the UKF/SPKF first and then to apply the GMSPPF or other HKPF if the computation power permits.

CONCLUSIONS

In this paper, we outlined three classes of nonlinear filtering techniques: (1) the EKF based on application of linear Gaussian KF to linearized process and measurement equations, (2) the UKF (SPKF) based on deterministic discrete approximation of Gaussian distributions and scaled unscented transformation (SUT) through nonlinear functions, and (3) the PF based on random samples representation of distribution and sequential importance sampling. In the latter case, to avoid sample depletion and degeneracy, the resampling and diversification steps are needed. To bring the transition prior-based proposal density close to the measurement likelihood function for importance weight updating, the hybrid Kalman particle filter was presented,

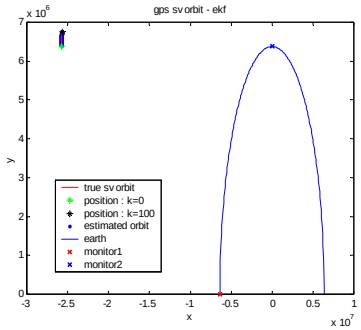

Figure 6 - Orbit Estimated by EKF

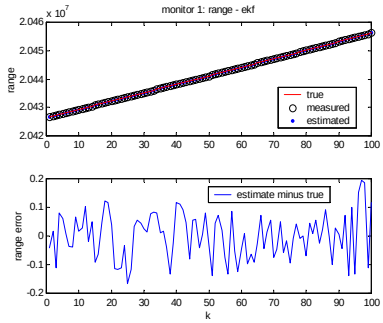

Figure 7 – Monitor 1's Measurements for EKF

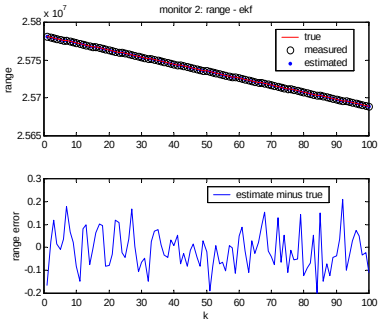

Figure 8 – Monitor 2's Measurements for EKF

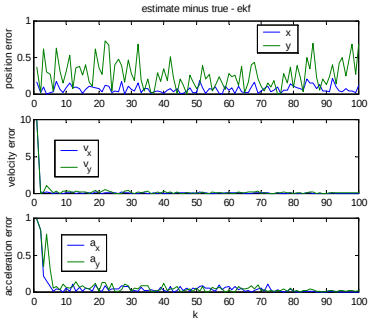

Figure 9 – State Estimation Errors for KF

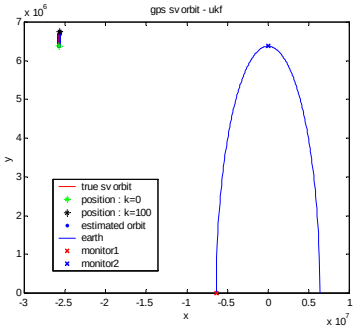

Figure 10 – Orbit Estimated by UKF

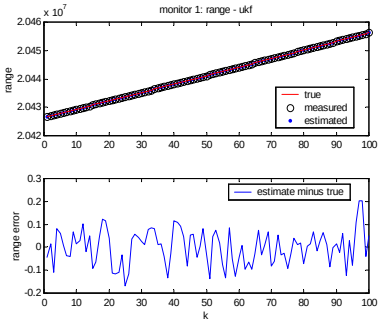

Figure 11 – Monitor 1's Measurements for UKF

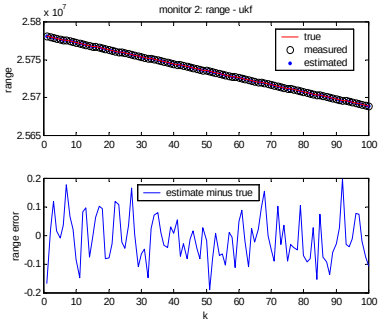

Figure 12 – Monitor 2's Measurements for UKF

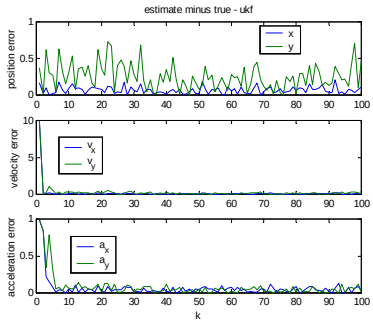

Figure 13 - State Estimation Errors for UKF

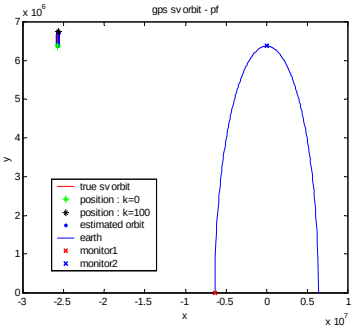

Figure 14 – Orbit Estimated by PF

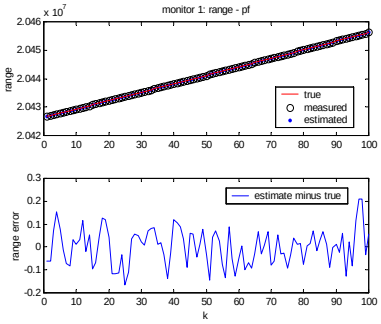

Figure 15 – Monitor 1's Measurements for PF

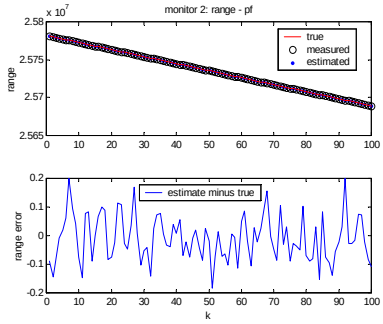

Figure 16 – Monitor 2's Measurements for PF

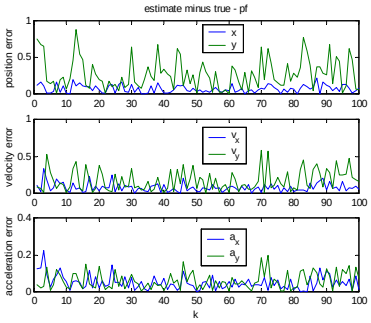

Figure 17 - State Estimation Errors for PF

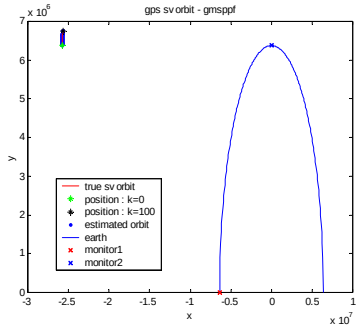

Figure 18 – Orbit Estimated by GMSPPF

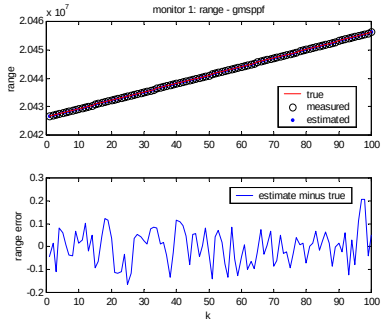

Figure 19 – Monitor 1's Measurements for GMSPPF

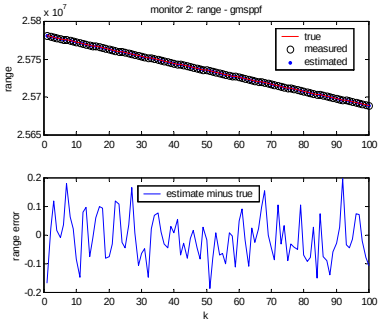

Figure 20 – Monitor 2's Measurements for GMSPPF

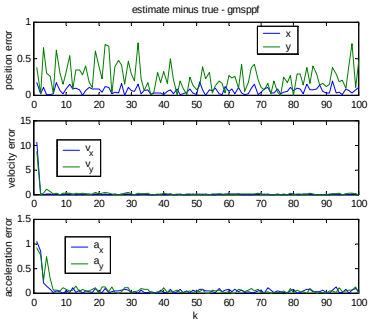

Figure 21 - State Estimation Errors for GMSPPF

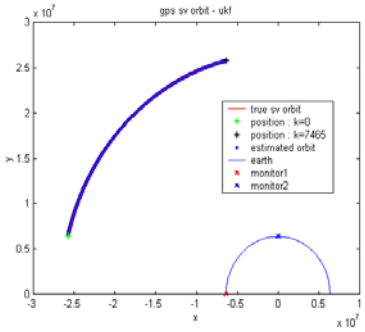

Figure 22 – Orbit Arc Visible to Two Monitor Stations

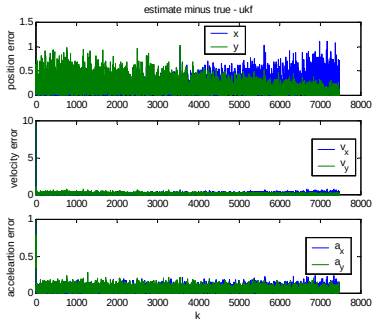

Figure 23 - State Estimation Errors over Long Orbit Arc

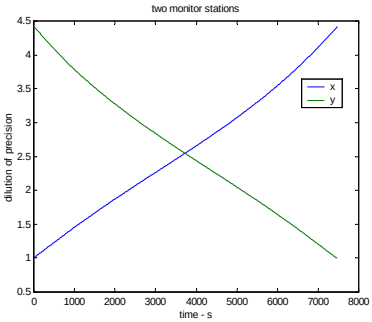

Figure 24 – GDOP for X- and Y-Components

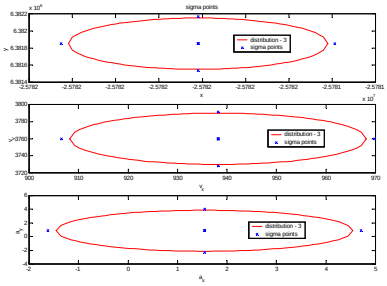

Figure 25 - Sigma-Points Selected for Position, Velocity, and Acceleration

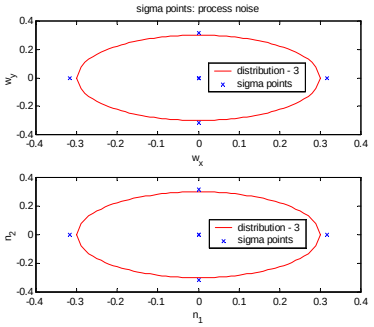

Figure 26 - Sigma Points Selected for Process Noise and Measurement Noise

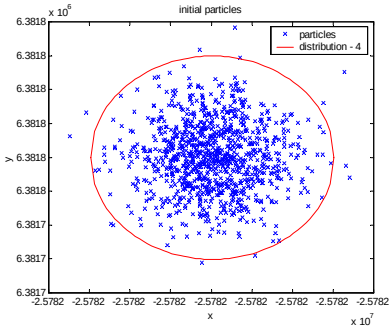

Figure 27 - Initial Particles Drawn from Estimated Distribution

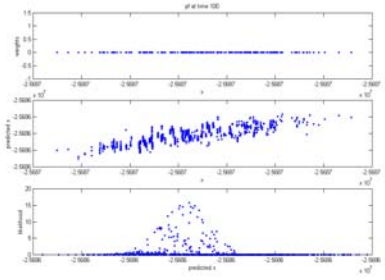

Figure 28 - Initial Weights, Predicted Particles, and Likelihood for X

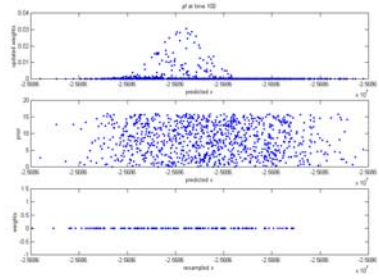

Figure 29 - Updated Weights, State Prior, and Weights After Resampling

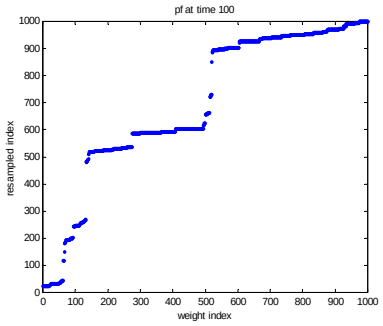

Figure 30 - Weight Index Before and After Resampling

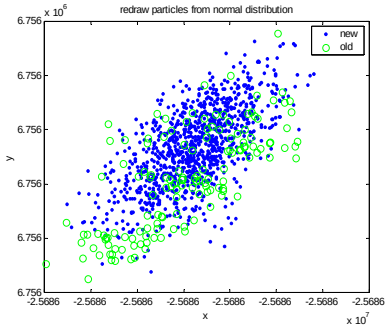

Figure 31 - Particle Redrawing with Estimated Distribution

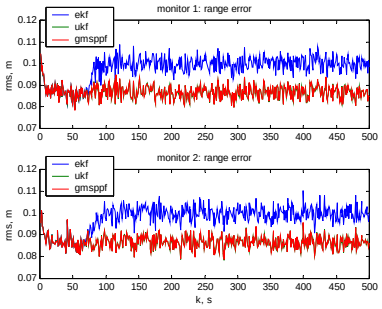

Figure 32 – RMS Values for Measurement Residuals

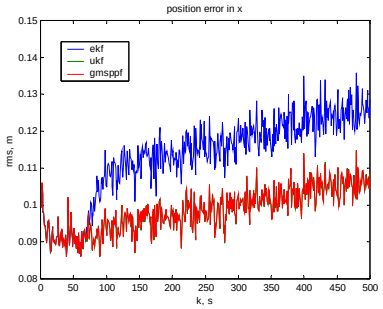

Figure 33 – RMS Values for Position Errors in X

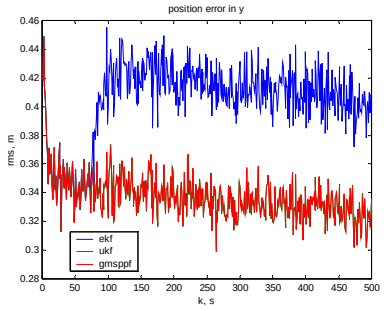

Figure 34 – RMS Values for Position Errors in Y

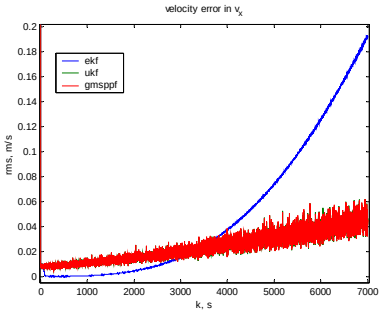

Figure 35 – RMS Values for Velocity Errors in X

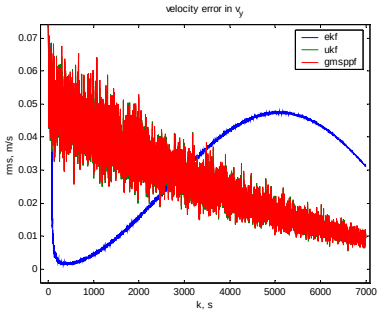

Figure 36 – RMS Values for Velocity Errors in Y

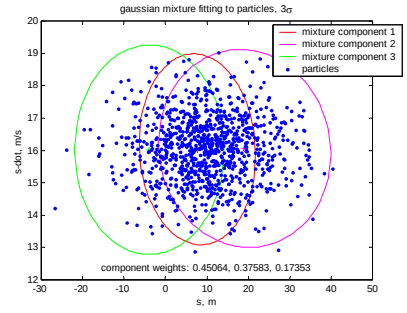

Figure 37 – Gaussian Mixture Fitting to Particles

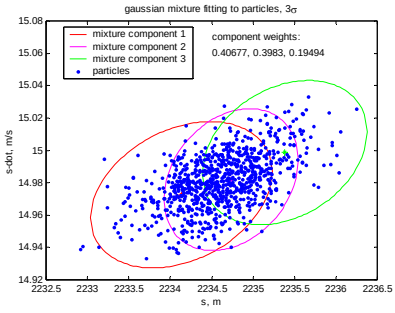

Figure 38 – Gaussian Mixture Fitting to Particles

in which an EKF or UKF (SPKF) is used to time-propagate each particle or each component of a Gaussian mixture fitted from all particles and measurement-update the predicted particle or Gaussian mixture component. The resulting posterior state distribution is used as the proposal distribution for importance sampling (IS)-based measurement updating in the second pass.

This paper also presented computer simulation results of four nonlinear filters (EKF, UKF/SPKF, PF, GMSPPF) applied to orbit determination using the kinematic method. Better results are expected for the dynamic method in orbit determination (study under way) where complicated nonlinearities are involved in various force models. Of particular interest in our current study is the use of a hybrid Kalman Particle Filter for ultra-tightly coupled GPS/IMU [Abbott, Lillo, and Douglas, 2000] using the grid-tracking scheme [Yang, 2003]. It is expected that these emerging nonlinear filtering techniques will gain more recognition in the navigation community and, in coupling with ever-increasing computing power, will benefit many navigation applications.

ACKNOWLEDGEMENT

Research reported in this paper was partially supported under Contract No. FA8650-04-M1633, which is gratefully acknowledged.

REFERENCES

A. Abbott, W. Lillo, and R. Douglas, "Ultra-Tight GPS/IMU Coupling Method," ION NTM, 2000.

P.O. Amblard, J.M. Brossier, and E. Moisan, "Phase Tracking: What Do We Gain from Optimality? Particle Filtering versus Phase-Locked Loops," *Signal Processing*, 83, 2002.

M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Trans. on Signal Processing*, 50(2), Feb. 2002.

B. Azimi-Sadjadi and P.S. Krishnaprasad, Integer Ambiguity Resolution in GPS Using Particle Filtering, *American Control Conf.*, June 2001

A. Doucet, N.J. J.F.G. de Freitas, N. Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2000.

P.M. Djuric, J.H. Kotecha, J.Q. Zhang, Y.F. Huang, T. Ghirmai, M.F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Signal Processing Magazine*, Sept. 2003.

S.G. Francisco, "GPS Operational Control Segment," in *Global Positioning System: Theory and Applications Volume I*, B.W. Parkinson and J.J. Spilker Jr. (eds.), AIAA, 1996.

P. Frykman, "*Applied Particle Filters in Integrated Aircraft Navigation*," Master Thesis, Linkopings Universitet, 2003

N.J. Gordon, D.J. Salmond, and A.F.M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings-F*, 140(2), April 1993.

N. Gordon, D. Salmond, and C. Ewing, "Bayesian State Estimation for Tracking and Guidance Using the Bootstrap Filter," *J. of Guidance, Control, and Dynamics*, 18(6), November-December 1995.

F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and J.J. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," *IEEE Trans. on Signal Processing*, 50(2), Feb. 2002.

S. Haykin (Ed.), *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., 2001.

R.A. Iltis, "Joint Estimation of PN Code Delay and Multipath Using the Extended Kalman Filter," *IEEE Trans. Communication*, 38(10), Oct. 1990.

S.J. Julier, J.K. Uhlmann, and H. Durrant-Whyte, A New Approach for Filtering Nonlinear Systems, *American Control Conf.*, 1995.

S.J. Julier, "The Scaled Unscented Transformation," *American Control Conf.*, 2002.

S.J. Julier and J.K. Uhlmann, "Unscented Filtering and Nonlinear Estimation," *Proc. of the IEEE*, 92(3), 2004.

E. Kraft, "A Quaternion-Based Unscented Kalman Filter for Orientation Tracking," *Int. Conf. on Information Fusion*, 2003.

J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer, 2001.

P. Maybeck, *Stochastic Models, Estimation and Control*, Academic Press, 1982.

D.T. Pham, K. Dahia, and C. Musso, A Kalman-Particle Kernel Filter and Its Application to Terrain Navigation, *Int. Conf. on Information Fusion*, 2003.

B. Ristic, S. Arullampalam and N. Gordon, *Beyond the Kalman Filter – Particle Filters for Trcaking Applications*, Artech House, 2004.

J.J.K. Ruanaidh and W.J. Fitzgerald, *Numerical Bayesian Methods Applied to Signal Processing*, Springer, 1996.

R. van der Merwe, A. Doucet, N. De Freita, and E. Wan, *The Unscented Particle Filter*, Technical Report CUED/F-INFENG/TR380, Cambridge University, 2000.

R. van der Merwe, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, Ph.D. Dissertation, Oregon Health & Science University, 2004.

C. Yang, "GPS Signal Tracking with Kalman Filter Based on Joint Code Delay and Carrier Phase and Frequency Error Discriminator," *ION AM*, June 2004

C. Yang, *Unscented Particle Filter for GPS Control Segment Precision Estimation*, Interim Report to AFRL/SNRP (FA8650-04-M-1633), August 2004.